

1. Comprehensive Introduction to Business Rules

Business rules are important to business professionals. The term business rules and “rules” and “assertions” as I am using them are all synonyms.

There are different camps with different views regarding how to implement business rules and leverage those rules to perform work for accounting professionals and other business professionals. There are the entrenched relational database software vendors that, over the years, have adapted relational database technology to meet the needs of their customers. Relational database systems offer proven, safe, reliable, predictable solutions. Then there is the semantic web camp. Their mantra is “anyone can say anything about anything” and with very flexible and powerful global standard technologies such as RDF and OWL 2 DL. There is the XBRL community with their offerings. There are niche worlds of PROLOG and DATALOG. Then there is the business rules management system group. Then there is the forty years of learning that the artificial intelligence community brings to the table.

Each camp says that *their answer* is the *best solution* to every problem. What is the right answer? Does there even need to be only one right answer?

The information in this document helps you sort through these changes, understand the importance of business rules, and cut through the hype and misinformation that tends to exist about new technologies.

1.1. Understanding what Business Rules Do

The Merriam-Webster dictionary defines anarchy¹ as “a situation of confusion and wild behavior in which the people in a country, group, organization, etc., are not controlled by rules or laws.” Business rules prevent information anarchy².

Business rules guide, control, suggest, or influence behavior. Business rules cause things to happen, prevent things from happening, or suggest that it might be a good idea if something did or did not happen. Business rules help shape judgment, help make decisions, help evaluate, help shape behavior, and help reach conclusions.

Business rules arise from the best practices of knowledgeable business professionals. A business rule is a rule that describes, defines, guides, controls, suggests, influences or otherwise constrains some aspect of knowledge or structure within some problem domain.

Don't make the mistake of thinking that business rules are completely inflexible and that you cannot break rules. Sure, maybe there are some rules that can never be broken. Maybe there are some rules that you can break. It helps to think of breaking rules as penalties in a football game. The point is that the guidance, control, suggestions, and influence offered by business rules is a choice of business

¹ Anarchy definition, Merriam-Webster, <http://www.merriam-webster.com/dictionary/anarchy>

² *Understanding that Business Rules Prevent Anarchy*, <http://xbrl.squarespace.com/journal/2016/7/15/understanding-that-business-rules-prevent-anarchy.html>

professionals. The meaning of a business rule is separate from the level of enforcement someone might apply to the rule.

A rule states a fact about the world (declarative rule). A rule can provide instructions (production rule).

1.1.1. Human-readable and machine-readable business rules

Business professionals interact with facts and rules every day and may not even realize it. Most business rules are in human readable form. But business rules can be represented in both human-readable form and machine-readable form. With the move to digital, more and more business rules are being represented in both human readable form and more importantly machine-readable form. Machine-readable business rules help automate processes which have been manual in the past.

Rules and facts must be written in some formal language in order for computer software applications to reason using the rules and draw conclusions about facts. Such a computer software application is often called a rule engine.

1.1.2. Using business rules to influence software behavior

Key to making an expert system or intelligent software agent³ work is business rules⁴ of the domain being put into machine-readable form.

Artificial intelligence⁵ is a branch of computer science. There are many good descriptions of artificial intelligence⁶. Here is one good definition:

Artificial intelligence is the automation of activities that we associate with human thinking and activities such as decision making, problem solving, learning and so on.

Those trying to make artificial intelligence work over the past 40 or so years have had limited success. But that is changing. Both under estimating or over estimating the capabilities the computer software will be able to achieve can have catastrophic consequences.

Expert systems⁷ is a branch of artificial intelligence. The following is a definition of an expert system:

Expert systems are computer programs that are built to mimic human behavior and knowledge. The computer program performs tasks that would otherwise be performed by a human expert. A model of the expertise of a domain of knowledge of the best practitioners or experts is put into machine-readable form and the expert system reaches conclusions or takes actions based on that information.

Intelligent software agents⁸ are computer code written in a specific way. An **agent** is an entity capable of **sensing** the **state** of its **environment** and **acting** upon it

³ Wikipedia, *Intelligent Agent*, retrieved July 24, 2016; https://en.wikipedia.org/wiki/Intelligent_agent

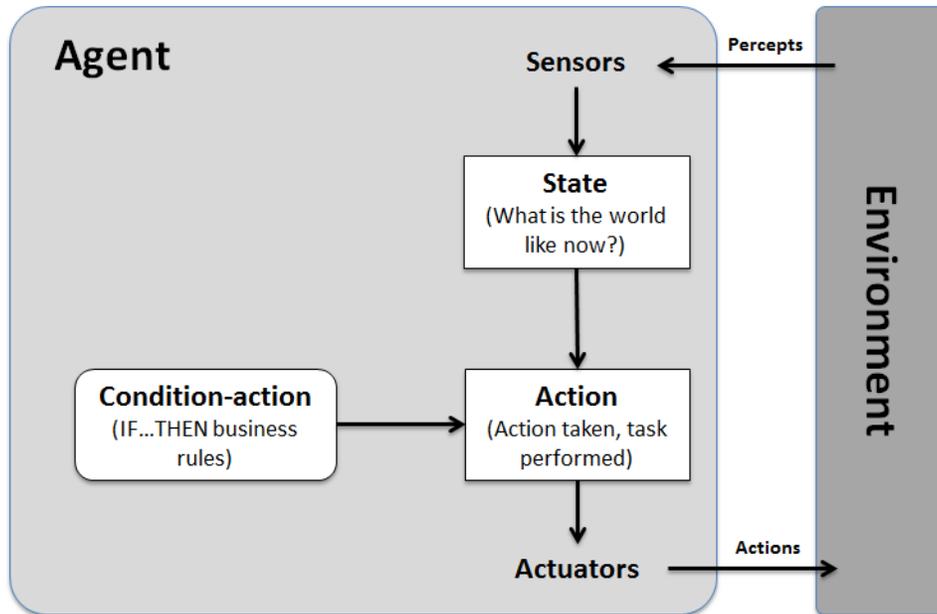
⁴ *Business Rules Manifesto*, <http://www.businessrulesgroup.org/brmanifesto.htm>

⁵ *Introduction to Artificial Intelligence Terminology*, <http://xbrl.squarespace.com/journal/2016/7/21/introduction-to-artificial-intelligence-terminology.html>

⁶ AlanTuring.net, *What is Artificial Intelligence?*, http://www.alanturing.net/turing_archive/pages/reference%20articles/What%20is%20AI.html

⁷ *Understanding the Components of an Expert System*, <http://xbrl.squarespace.com/journal/2016/5/24/understanding-the-components-of-an-expert-system.html>

based on a **set of specified rules**. An agent performs specific tasks on behalf of another. In the case of software, an agent is a software program. Consider that definition of an agent and look at the graphic below to get an idea of how an intelligent agent software works:



An intelligent agent is software that assists people and acts on their behalf. Intelligent agents work by allowing people to:

- delegate work that they could have done to the agent software,
- perform repetitive tasks,
- remember things you forgot,
- intelligently find, filter and summarize complex information,
- customize information to your preferences,
- learn from you and even make recommendations to you.

1.1.3. Business rules drive intelligent software agents and expert systems automating work

The global consultancy firm Gartner classifies XBRL as a transformational technology⁹. Gartner defines transformational as something that "*enables new ways of doing business across industries that will result in major shifts in industry dynamics*". Major shifts mean lots of change and some winners and some losers.

An example of one major shift is provided by what professional accountants call the "disclosure checklist". Accountants creating financial reports often use accounting and reporting checklists or "disclosure checklists" as memory joggers to help them

⁸ *Introduction to Intelligent Software Agents for Business Professionals*, http://xbrl.azurewebsites.net/2016/Library/02_IntroducingIntelligentAgents.pdf

⁹ Charles Hoffman and Liv Watson, *XBRL for Dummies*, page 145

get the reports right¹⁰. These memory joggers were created to be read by humans and can be a couple hundred pages. What if a financial report was structured, such as an XBRL-based public company financial report that must be submitted to the U.S. Securities and Exchange Commission? What if these human-readable memory joggers could be made machine-readable? And what if an intelligent software agent could be created to automate the manual task of checking a financial report to make sure that report was mechanically correct.

Note the statement “mechanically correct”. This is a very important distinction. No computer program will ever have the judgement of a professional accountant. See the section *Setting the right expectations* later in this document. But computer programs can perform work if the financial report is structured and the necessary business rules are made machine-readable. How much of a disclosure checklist can be automated? That percentage is as-of-yet to be determined. Perhaps 20% can be automated or even 80% will be automated. Maybe even a higher percentage. The probability 0% of a disclosure checklist can be automated is extremely low.

Humans augmented by machine capabilities, much like an electronic calculator enabling a human to do math quicker, will empower knowledge workers who know how to leverage the use of those machines.

1.1.4. Business rules are metadata and follow the rules of formal logic

Business rules provide a thick metadata layer that enables computer systems to perform useful work. The more business rules that exist, the more work a computer system can perform.

A simple example of a business rule the accounting equation¹¹ which is universally accepted: *Assets = Liabilities and Equity*. The accounting equation is an axiom. *Axioms* describe self-evident logical principles that no one would argue with. The accounting equation is the foundation upon which double-entry accounting is built. *Theorems* are deductions which can be proven by constructing a chain of reasoning by applying axioms in the form of IF...THEN statements. These axioms and theorems are the basis for formal logic¹². Formal logic is a discipline of philosophy which has been around since the days of Aristotle. Computers work based on formal logic. But you have to be careful. Computers are dumb beasts (see the section *Major obstacles to harnessing the power of computers*). Computers cannot follow all the rules of logic, only a limited set. To effectively get the tool, the computer, to perform work correctly, you need to follow certain guidelines and not push the tool beyond its capabilities.

1.1.5. Basic example of business rules, fundamental accounting concept relations

A basic example of business rules is the fundamental accounting concept relations¹³. The fundamental accounting concept relations build upon the accounting equation.

¹⁰ *Automating Accounting and Reporting Checklists*, <http://xbri.squarespace.com/journal/2016/5/5/automating-accounting-and-reporting-checklists.html>

¹¹ Wikipedia, *Accounting Equation*, retrieved August 3, 2016, https://en.wikipedia.org/wiki/Accounting_equation

¹² Wikipedia, *Logic*, retrieved August 3, 2016, <https://en.wikipedia.org/wiki/Logic>

¹³ *Public Company Quality Continues to Improve, 84% are Consistent*, <http://xbri.squarespace.com/journal/2016/7/1/public-company-quality-continues-to-improve-84-are-consisten.html>

For example, assets can be broken down into current and noncurrent portions. And so you can create the rule: Assets = Current assets + Noncurrent assets. Another rule is: Liabilities = Current liabilities + Noncurrent liabilities.

But accountants understand that not every economic entity breaks assets and liabilities down into their current and noncurrent portions. For example, banks report using an unclassified balance sheet, reporting only total assets and total liabilities.

To address this, another rule is created. Economic entities are partitioned into groups based on how they report, their reporting style. Those that provide classified balance sheets and therefore report current and noncurrent assets and liabilities are put into one group. Those that report using an unclassified balance sheet, reporting only totals for assets and liabilities, are put into a different group. And so the result is rules that are universally applicable to all economic entities. Not every economic entity uses every fundamental accounting concept relations rule, only rules that apply to the group that the economic entity is in are applicable. This is only a brief description of the fundamental accounting concept relations. See the actual fundamental accounting concept relations metadata¹⁴ for a more comprehensive explanation.

An example of a more comprehensive set of business rules is accounting and reporting checklists¹⁵ that are used by professional accountants to create external financial reports. Today, these business rules are organized in the form of a human-readable accounting and disclosure checklist which are used as a memory jogger by professional accountants creating a financial report.

1.1.6. Business rules can result in a theory

Business rules can result in a theory as to how the business domain covered by the business rules behaves. Looking at this from 180 degrees, the opposite perspective, if you have a set of business rules you can make computer software behave like the business domain.

For example, there are approximately 6,500 public companies that submit XBRL-based financial reports to the U.S. Securities and Exchange Commission. (Note that this explicitly excludes economic entities which are funds or trusts because they follow different rules which are not provided by this set of rules because I am not interested in funds or trusts.)

Those 6,500 economic entities can be broken down into 104 different reporting styles. About 80% of all economic entities fit into a set of only 12 different reporting styles, so there are some large groups and much smaller groups of reporting styles, some groups are comprised of just one economic entity.

As of the most current results, 99.88% of economic entities are consistent with each of the individual fundamental accounting concept relations applicable to that entity. A

¹⁴ *Fundamental Accounting Concept Relations*, <http://xbrl.squarespace.com/fundamental-accounting-concept/>

¹⁵ *Automating Accounting and Reporting Checklists*, <http://xbrl.squarespace.com/journal/2016/5/5/automating-accounting-and-reporting-checklists.html>

total of 88.8% of economic entities are consistent with all of the fundamental accounting concept relations for their specific reporting style¹⁶.

Different software vendors and filing agents are more consistent with the fundamental accounting concept relations than are others. A total of 6 software vendors/filing agents have 96% or more of all of their customers consistent with these basic, fundamental accounting concept relations. Whereas there are 9 software vendors/filing agents that have 75% or less of their customers consistent with these basic relations.

Does this mean that there is some correlation between what software vendor/filing agent that is used and the fundamental accounting concept relations that exist? No. What this information points out is that different software vendors/filing agents are better, or worse, and creating XBRL-based financial reports that are consistent with the basic, fundamental accounting concept relations.

So again, now look at this from the opposite perspective. The business rules articulated by the fundamental accounting concept relations can influence the software of software vendors/filing agents so that their financial reports created are consistent with the fundamental rules of financial reporting. Said succinctly, software can help professional accountants create better financial reports.

1.2. Business Rules and Business Professionals

Business professionals interact with business rules every day and may not even realize it. Most business rules are in human-readable form. But business rules can be represented in both human-readable form and machine-readable form. With the move to digital, more and more business rules are being represented in both human readable form and more importantly machine-readable form. Machine-readable business rules help automate processes which have been manual in the past.

1.2.1. Business professionals create and maintain business rules

Business professionals create and maintain business rules. Imagine an information technology professional being responsible for maintaining the fundamental accounting concept relations rules. That simply would never work. To maintain the fundamental accounting concept relations rules one needs to have an intimate understanding of financial accounting and financial reporting.

As Article 9 of the *Business Rules Manifesto*¹⁷ states, business rules are of, by, and for business people; not IT people.

- 9.1. Rules should arise from knowledgeable business people.
- 9.2. Business people should have tools available to help them formulate, validate, and manage rules.
- 9.3. Business people should have tools available to help them verify business rules against each other for consistency.

Rather than creating tools that only IT professionals can use because they are so complicated; business professionals need to demand software tools that properly

¹⁶ Quarterly XBRL-based Public Company Financial Report Quality Measurement (December 2018), <http://xbrl.squarespace.com/journal/2018/12/31/quarterly-xbrl-based-public-company-financial-report-quality.html>

¹⁷ *Business Rules Manifesto*, <http://www.businessrulesgroup.org/brmanifesto.htm>

expose functionality that exposes business rules to business users such that business users are working with business domain knowledge, not technical details that should have been buried deeply within the software applications.

Business professionals need to understand the *Law of Conservation of Complexity*¹⁸ which states that complexity can never be removed from a system, but complexity can be moved. The *Law of Conservation of Complexity* states: "Every application has an inherent amount of irreducible complexity. The only question is: Who will have to deal with it - the user, the application developer, or the platform developer?"

1.2.2. Reusing and sharing business rules

The *Business Rules Manifesto*, Article 4¹⁹, points out that business rules should be declarative rather than procedural. The declarative approach has important advantages including that your business rules become reusable across both processes and software platforms. As such, the rules become both highly re-engineerable and highly re-deployable.

Declarative involves stating *that* something is the case. Procedural involves stating *how* to do something. The following is a simple example of procedural rules and declarative: Suppose you desire a cup of coffee.

Procedural:

1. Go to kitchen.
2. Get water, coffee, sugar, cream.
3. Heat the water on the stove until the water boils.
4. Put the coffee, sugar, and cream into the water.
5. Bring the result to me.

Declarative:

1. Get me a cup of coffee.

Taking a procedural approach you define the entire process and provide each step necessary to obtain the desired result. Taking a declarative approach you state the desired result, and let the system determine the best way to get that result; all you care about is the result without worrying how the result will be achieved.

A procedure is used in only one way, but a declarative specification can be used in many different ways²⁰. Again, either a procedural or a declarative approach to representing business rules will work. It is important to understand the pros and cons of each approach.

Next, business rules should not be mixed within software application code. Why? There are three reasons. First, if business rules are within application code then it takes a programmer to change the code. Second, if the business rules are

¹⁸ Understanding the Law of Conservation of Complexity, <http://xbrl.squarespace.com/journal/2015/5/24/understanding-the-law-of-conservation-of-complexity.html>

¹⁹ *Business Rules Manifesto, Article 4. Declarative, Not Procedural*, <http://www.businessrulesgroup.org/brmanifesto.htm>

²⁰ John F. Sowa, *Fads and Fallacies about Logic*, page 3, <http://www.jfsowa.com/pubs/fflogic.pdf>

embedded within one software application that it is challenging to reuse those same rules within another application. Third, sharing business rules becomes easy.

Think of what it would mean if you could create a spreadsheet and test the spreadsheet against a shared set of rules provided somewhere on the Internet or on a private intranet.

This does not mean that all business rules become publically available. Using security rules could be available only to a department, or a company, or even across an entire supply chain.

Basically, just as applications and the databases they use were separated from one another (they were combined in the early days of software); business rules and applications are now being separated.

1.2.3. Example of business rule reuse

An example of business rule reuse can be seen via the fundamental accounting concept relations rules^{21,22}. All the metadata for these rules are declarative in nature and represented using the XBRL global standard. One software vendor, XBRL Cloud, uses the fundamental accounting concept relations rules to validate XBRL-based financial filings which are submitted to the SEC²³. Another software vendor uses the exact same metadata and completely different software to query XBRL-based financial filings^{24,25}.

1.3. Examples of Types of Business Rules

The best way to understand what business rules are is to have some examples of business rules. Business rules can be put into general groups by type of rule. Different people group business rules into different groups such as "structural rules" or "behavioral rules". Others break them down into "quality logic" and "business logic".

1.3.1. General types of business rules

At their essence, business rules articulate information about something or about the relationship between one thing and some other thing. Some examples that can help you better understand exactly what they are:

- **Assertions:** For example asserting that the balance sheet balances or "Assets = Liabilities + Equity".

²¹ *Fundamental Accounting Concept Relations*, <http://www.xbrlsite.com/2016/fac/v3/Documentation/#home>

²² *Crash Course in the Fundamental Accounting Concepts*, <http://www.xbrlsite.com/2016/Prototype/FundamentalAccountingConceptsPureXBRL/Documentation/FundamentalAccountingConcepts.html>

²³ *XBRL Cloud's Validation Report*, <http://xbrlsite.azurewebsites.net/2016/ReferenceImplementation/2016-04-15/evidence-package/index.html#USFACRenderingSummary.html>

²⁴ *Reporting Style Examples*, <http://www.xbrlsite.com/2016/fac/v3/Examples/Index.html>

²⁵ *Comparison Formats*, http://www.xbrlsite.com/2016/fac/v3/Examples/Index_Compare.html

- **Computations:** For example, calculating things, such as "Total Property, Plant and Equipment = Land + Buildings + Fixtures + IT Equipment + Other Property, Plant, and Equipment".
- **Constraints:** For example, specific behavioral constraints that control when it is appropriate to create, update, or remove information.
- **Continuity cross-checks:** For example, if a fact is used that fact does not conflict with or contradict other facts.
- **Process-oriented rules:** For example, the disclosure checklist commonly used to create a financial statement which might have a rule, "If Property, Plant, and Equipment exists, then a Property, Plant and Equipment policies and disclosures must exist."
- **Regulations:** Another type of rule is a regulation which must be complied with, such as "The following is the set of ten things that must be reported if you have Property, Plant and Equipment on your balance sheet: depreciation method by class, useful life by class, amount under capital leases by class ..." and so on. Many people refer to these as reportability rules.
- **Instructions or documentation:** Rules can document relations or provide instructions, such as "Cash flow types must be either operating, financing, or investing."
- **Relations:** How things can be related, such as whole-part relations. For example, how the business segments of an economic entity are related.

1.3.2. Categories of business rules from the XBRL perspective

Business rules can be categorized. The Business Rules Group provides a good summary of business rule categories. But sometimes how terms are used differ between groups using such terms. Below I have provided a set of business rule categories inspired by the Business Rules Group categories which is then reconciled to XBRL terminology.

- **Definition of business terms:** The very definition of business terms is a category of business rule. Each term is a rule. In XBRL, the report elements defined in an XBRL taxonomy schema is how business terms are defined. Terms are essentially identifiers. In XBRL, terms are grouped into one of the following categories of terms: Network, Hypercube (a.k.a. Table), Dimension (a.k.a. Axis), Member, Primary Items (a.k.a. Line Items), Primary Item (a.k.a. a concrete Concept), and Abstract (a.k.a. an abstract Concept or Primary Item). Business professionals have to go through a process of naming things that exist in reality and giving them names as contrast to providing additional preferred labels for names that already exist. These terms describe how business professionals think and talk about real world notions, ideas, and other such phenomenon. The definitions of terms in the past have been documented in the form of human-readable glossaries. We now make these terms human-readable and machine-readable by defining them in XBRL taxonomy schemas.
- **Structural assertions:** This term appears to describe two types of structures in XBRL: what XBRL calls a "fact" and what XBRL calls "relations" (presentation, calculation, definition, XBRL formula). Structural assertions can be documented as natural language sentences or described graphically

has hierarchies as relationships, qualities, and other such structures. There are two important distinct types of structures in XBRL:

- **XBRL Fact:** A fact in XBRL is something that is reported within an XBRL instance. A fact is a structure comprised of other structures generally defined in the form of terms in an XBRL taxonomy schema but there are a few things defined in the XBRL instance itself (entity identifier, period, XBRL footnotes). So, a fact is a hard-coded structural assertion defined by the XBRL technical specification. A fact has an aspect model. This is the same as what I call the multidimensional model of XBRL.
- **Other XBRL relations:** This category of structural assertions includes all other relations definable using XBRL including presentation, calculation, definition, and XBRL formula relations.
- **Action assertions:** Action assertions constrain or influence behavior in some way. Action assertions cause things to happen or prevent things from happening. They can also be used to make suggestions. XBRL Formula provides for existence assertions, consistency assertions, and value assertions.
- **Derivations:** A derivation is a mathematical algorithm or a logical inference (induction or deduction) that is used to derive, or what I have called impute, other structural relations (i.e. XBRL facts or other relations). Derivations create new knowledge based on existing knowledge. XBRL Formula has a mechanism for creating new facts.

A bit of clarification is helpful to make sure all of the above is clear. The notion of derivations might not be familiar to some people or you might be familiar with it in different terms. Here is some clarifying information that distinguishes between explicitly provided facts and derived facts:

- **Base Fact:** a base fact is a fact that has been explicitly reported in a financial report. For example, if you report the fact "us-gaap:Assets" for a specific economic entity for a specific period.
- **Derived Fact:** a derived fact is a fact whose value is created by an inference or mathematical computation. For example, if the base facts "us-gaap:Assets" and "us-gaap:AssetsCurrent" are reported then the fact "us-gaap:AssetsNoncurrent" can be derived because of two pieces of information: (1) the values of us-gaap:Assets and us-gaap:AssetsCurrent are known and (2) the business rule "Assets = Current assets + Noncurrent assets" is known; so deductive reasoning can be used to obtain the derived fact "us-gaap:AssetsNoncurrent".
- **Derivation:** a derivation is an algorithm used to infer or compute a Derived Fact. (i.e. a business rule). In the derived fact example above, the derivation is "Assets = Current assets + Noncurrent assets". There are two types of derivations:
 - **Logical inference:** a logical inference is a Derivation that produces a Derived Fact using logical induction (from particulars) or deduction (from general principles).
 - **Mathematical inference:** a Derivation that produces a Derived Fact according to a specified mathematical algorithm.

1.3.3. Logical layers of business rules

Business rules can be organized into convenient logical layers that interact with one another. These layers²⁶ contribute to helping business professionals manage, use, and otherwise organize and interact with business rules.

- **Flow logic:** (sequence, process or flow)
 - **Procedural logic** – model sequence, loop, or iterative procedures.
 - **Flow logic** – fully automated sequence of operations, actions, tasks, decisions, rules.
 - **Workflow logic** – type of flow logic, semi-automated or manual processes that need an action to be taken from outside the system by another system or human.
- **Information assertions:** (compliance, quality, consistency, completeness, accuracy)
 - **Business terminology logic** – definition of business concepts
 - **Validation logic** – validate action assertions.
 - Action assertions
 - Structural assertions
 - **Decision logic** – type of validation logic, handles execution que and conflict resolution
 - Execution logic
 - Conflict resolution logic
 - **Derivation logic** – deviations which derives new facts using existing facts, rules, and logical or mathematical reasoning
 - Logical inference
 - Mathematical inference

1.4. Understanding the Business Rules Engine

Rather than a computer programmer writing a bunch of IF...THEN sequential computer code to enforce business rules as had been the case in the past, today business rules are enforced differently. There are two pieces to enforcing business rules:

1. Writing the business rules
2. Using a business rules engine to process business rules

This separation of concerns produces many significant benefits. Business rules can be repurposed for many things, flexibility, easier maintenance, and so forth. So what is a business rules engine? What type of business rules engine do you need?

- **Semantic reasoner**²⁷: A semantic reasoner, reasoning engine, rules engine, or simply a reasoner, is a piece of software able to infer logical consequences

²⁶ This is one of the better explanations of these different layers that I have run across, <http://wiki.flexrule.com/index.php?title=Logic>

from a set of asserted facts or axioms. The notion of a semantic reasoner generalizes that of an inference engine, by providing a richer set of mechanisms to work with.

- **Inference engine**²⁸: The inference engine applied logical rules to the knowledge base and deduced new knowledge.
- **Business rules engine**²⁹: A business rules engine is a software system that executes one or more business rules in a runtime production environment. Some problem solving logic³⁰ and some problem solving method is used by the business rules processor.

1.4.1. Business rules engine terminology

This blog post³¹ about semantic reasoners provides the following definition which points out that there are many different terms that refer to what might be, or might not be, the same sort of thing:

A semantic reasoner, reasoning engine, rules engine, or simply a reasoner, is a piece of software able to **infer logical consequences from a set of asserted facts or axioms**. The notion of a semantic reasoner generalizes that of an inference engine, by providing a richer set of mechanisms to work with. The inference rules are commonly specified by means of an ontology language, and often a description language. Many reasoners use first-order predicate logic to perform reasoning; inference commonly proceeds by forward chaining and backward chaining.

Breaking these terms out, these are the important pieces to understand:

- **Machine or engine**: an apparatus using or applying mechanical power and having several parts, each with a *definite function* and together performing a *particular task*.
- **Inference**: a business rules engine either does, or does not, include the ability to infer new information using existing information and logical rules.
- **First-order predicate logic**: Formal logic was consciously broken into two groups: first-order logic and higher-order logic. There is a reason for this. Systems based on first-order logic can be proven to be sound (all provable theory statements are true in all models) and complete (all theory statements which are true in all models are provable using proof theory). Higher-order logics are less well-behaved than those of first-order logic. They are less predictable and therefore less reliable and they are significantly harder to implement using computers. That is why computer systems are generally based on first-order logic.

²⁷ Wikipedia, *Semantic Reasoner*, retrieved August 2, 2016, https://en.wikipedia.org/wiki/Semantic_reasoner

²⁸ Wikipedia, *Inference Engine*, retrieved August 2, 2016, https://en.wikipedia.org/wiki/Inference_engine

²⁹ Wikipedia, *Business rules engine*, retrieved August 2, 2016, https://en.wikipedia.org/wiki/Business_rules_engine

³⁰ *Comprehensive Introduction to the Notion of Problem Solving Logic for Professional Accountants*, <http://xbrlsite.azurewebsites.net/2016/Library/ComprehensiveIntroductionToNotionOfProblemSolvingLogicForProfessionalAccountants.pdf>

³¹ Hello Semantic Web, *Semantic Reasoners*, retrieved August 2, 2016, <http://hellosemanticweb.blogspot.com/2011/04/semantic-reasoners.html#axzz2URUuQy00>

- **Logical catastrophes**³²: There are things that someone can do under first-order predicate logic that cause what I refer to as logical catastrophes or system failure points. For example, inadvertently putting a system into an infinite loop from which it cannot escape is a logical catastrophe. The possibility of all such catastrophes must be eliminated from business rules systems.
- **Horn Clauses**: Horn Clauses is a safe subset of first-order predicate logic. PROLOG is limited by Horn Clauses. However, PROLOG still has specific cases which cause systems to inadvertently break. As such, DATALOG was created to create an even safer set of first-order predicate logic.
- **Multidimensional model**: Transaction processing systems (OLTP³³) and analytical systems (OLAP³⁴) are used for different things. OLAP leverages a multidimensional model which makes querying information more flexible and efficient. Business rules engines might need to have an inherent understanding of the multidimensional model³⁵. XBRL provides a global standard multidimensional model³⁶. RDF also provides a global standard dimensional model³⁷.
- **Mathematics**: Business rules engines or semantic processors need to have an inherent understanding of mathematics to the extent that mathematics is used within business rules.
- **Standard or proprietary**: There are standard approaches to implementing rules engines and there are proprietary approaches and tools. Some business rules engines might be open source.
- **XBRL Formula processor**: An XBRL formula processor is a type of business rules engine. There is a global standard specification³⁸ for expressing business rules using XBRL.

1.4.2. Best business rules engines

So, what is the best business rules engine? There are different categories of semantic reasoners:

- **OWL reasoners**³⁹: Business rules engines, or reasoners, that work with OWL. https://en.wikipedia.org/wiki/Semantic_Web_Stack

³² *Brainstorming the Idea of Logical Catastrophes or Failure Points*, <http://xbrl.squarespace.com/journal/2015/7/25/brainstorming-idea-of-logical-catastrophes-or-failure-points.html>

³³ Wikipedia, *Online Transaction Processing*, Retrieved August 3, 2016, https://en.wikipedia.org/wiki/Online_transaction_processing

³⁴ Wikipedia, *Online Analytical Processing*, https://en.wikipedia.org/wiki/Online_analytical_processing

³⁵ *Introduction to the Multidimensional Model for Professional Accountants*, <http://xbrl.squarespace.com/journal/2016/3/18/introduction-to-the-multidimensional-model-for-professional.html>

³⁶ *XBRL Dimensions 1.0*, <https://specifications.xbrl.org/work-product-index-group-dimensions-dimensions.html>

³⁷ *The RDF Data Cube Vocabulary*, <http://www.w3.org/TR/vocab-data-cube/> also see <https://dvcs.w3.org/hg/qld/raw-file/29a3dd6dc12c/data-cube/index.html>

³⁸ *XBRL Formula Specification*, <https://specifications.xbrl.org/work-product-index-formula-formula-1.0.html>

³⁹ W3C, *OWL Reasoners*, <https://www.w3.org/2001/sw/wiki/OWL/Implementations>

- **List of OWL Reasoners:** <http://owl.cs.manchester.ac.uk/tools/list-of-reasoners/>
- **Free open source reasoners:** https://en.wikipedia.org/wiki/Semantic_reasoner#Free_Software_.28Open_Source.29
- **XBRL Formula processors:** Business rules engines that work with XBRL Formula. For example Arelle (<http://arelle.org/>) is XBRL software that includes an XBRL Formula processor.
- **Proprietary rules engines:** Business rules processors can use global standard and/or proprietary information and rule formats. Here are a few such business rules engines:
 - FlexRule: <http://www.flexrule.com/>
 - Fluent Editor: <http://www.cognitum.eu/semantics/FluentEditor/>
 - Ergo Suite: <http://coherentknowledge.com/product-overview-ergo-suite-platform/>
- **SBVR processors**⁴⁰: SBVR processors are conformant to the Semantics of Business Vocabulary and Business Rules (SBVR) OMG standard.
- **Other rules engines:** This is a list of some other interesting rules engines.
 - **Jess:** <http://www.jessrules.com/jess/index.shtml>
 - **Clips:** <http://clipsrules.sourceforge.net/>
 - **Prolog:** <http://www.swi-prolog.org/>
 - **Datalog:** <http://cs.nyu.edu/faculty/davise/ai/datalog.html>
 - **Datomic:** <http://docs.datomic.com/query.html>
 - **Clojure:** <https://clojure.org/>
 - **Drools:** <https://www.drools.org/>
 - **Easy rules:** <http://www.easyrules.org/>
 - **InfoQ:** <https://www.infoq.com/articles/Rule-Engines>
 - **BizTalk:** <https://www.microsoft.com/en-us/cloud-platform/biztalk>
 - **JRules:** <https://www.ibm.com/cloud-computing/products/business-process-management/business-rules/>
 - **OpenRules:** <http://openrules.com/>
 - **Java rules engine:** <https://jcp.org/en/jsr/detail?id=94>; https://en.wikipedia.org/wiki/JSR_94
 - **XSB:** <http://xsb.sourceforge.net/>
 - **FLORA:** <http://flora.sourceforge.net/>

⁴⁰ OMG, *Semantics of Business Vocabulary and Business Rules (SBVR)*, section 2.5 Conformance of an SBVR Processor, page 7, <http://www.omg.org/spec/SBVR/1.0/>

- **SILK:** <http://silkframework.org/>;
<http://web.stanford.edu/class/cs227/Readings/20100426-SILK-specification.pdf>

Some business rules engines can be considered GOOD. Others can be considered BETTER. And even others can be considered BEST. The following is a comparison and gap analysis of different software that provides business rules engine type functionality⁴¹:

Business Rules Processor	For more information	Information System/Format	XBRL Processor	XBRL Formula Processor	Supports Inference	Problem Solving Method	Supports Probabilistic Reasoning	State Machine	General Problem Solving Logic Capabilities	Fact Database	Knowledge Base	Rule Creation Interface (XBRL Taxonomy, XBRL Formulas, Other)	Understanding of Business Report Conceptual Model	Understanding of Financial Report Conceptual Model	Justification and Explanation Mechanism (Transparency into Use of Reasoning and Information Provenance)	Knowledge Acquisition Mechanism
Arelle (Open source API level interface)	http://arelle.org/	Standard XBRL	Yes	Yes	No	Sequential	No	No	INCOMPLETE, Limited to XBRL Formula	XBRL Instance	XBRL Taxonomy	None provided	No	No	Not provided, can be created	Manual
XBRL Development Tools (Above)	https://www.altova.com/xbrl-tools	Standard XBRL	Yes	Yes	No	Sequential	No	No	INCOMPLETE, Limited to XBRL Formula	Raptor/XML+XBRL Server	Raptor/XML+XBRL Server	XBRL Specific but oriented to technical users	No	No	Not provided, can be created	Manual
Sphix (Corefiling)	https://www.corefiling.com/products/sphix/	Standard XBRL	Yes	Yes	No	Sequential	No	No	INCOMPLETE, Limited to XBRL Formula	XBRL Instance	XBRL Taxonomy	XBRL Specific but oriented to technical users	No	No	Unknown	Manual
Clean Score (XBRL Cloud)	https://www.abricloud.com/clean-score-a.html	Profile based Standard XBRL	Yes	Yes	Yes	Sequential	No	No	GOOD (Subset of RuleLog)	XML Infonet stored in file system	XML Infonet stored in file system	None provided	Yes	Yes	Good, usable by business professionals	Manual
PepperNet Knowledge Based Financial Report Creation System	http://pepper.net.azurewebsites.net/	Profile based Standard XBRL	Yes	No	Yes	Forward chaining	No	Yes	GOOD (Subset of RuleLog)	XML Infonet stored in file system	XML Infonet stored in file system	None provided at present time, will be business user oriented	Yes	Yes	Good, usable by business professionals	Manual
FlexRule Business Logic Platform	http://www.flexrule.com/solutions/	Proprietary or general format	No	No	Yes	Forward chaining	No	No	BETTER (larger subset of RuleLog)	Proprietary or general format	Proprietary or general format	Comprehensive but oriented to technical users	No	No	Unknown	Manual
InRule (InRule Technologies)	http://www.inrule.com/products/inrule/	Proprietary or general format	No	No	Yes	Forward chaining	No	No	BETTER (larger subset of RuleLog)	Proprietary format	Proprietary format	Oriented toward non-technical users and business professionals	No	No	Unknown	Manual
Smarts (Sparkling Logic)	https://www.sparklinglogic.com/insight/decision-manager/	Proprietary or general format	No	No	Yes	Forward chaining	No	No	BETTER (larger subset of RuleLog)	Proprietary or general format	Proprietary or general format	Comprehensive but oriented to technical users	No	No	Unknown	Manual
Fluent Editor (Capitum)	http://www.capitum.eu/semantic/rule-engine/	Standard RDF, OWL, SPIN	No	No	Yes	Forward chaining	No	No	BETTER (larger subset of RuleLog)	RDF stored in file system	RDF, OWL, SPIN stored in file system (Semantic Web Stack)	Comprehensive but oriented to technical users	No	No	Unknown	Manual
Topraid Platform (TopQuadrant)	https://www.topquadrant.com/techology/topraid-platform-overview/	Standard RDF, SHACL, RDFS, OWL, SPIN	No	No	Yes	Forward and Backward chaining	Yes	No	BEST (RuleLog plus)	RDF triple store repository	SHACL, RDFS, OWL, SPIN stored in file system (Semantic Web Stack)	Comprehensive but oriented to technical users; rule creation templates usable by business	No	No	Good, usable by business professionals	Manual or Automated
Ergo Logic System (Coherent Knowledge)	http://coherentknowledge.com/financial-domain-application/	Standard RDF, RIF	No	No	Yes	Forward and Backward chaining	Yes	No	BEST (RuleLog plus)	RDF stored in file system	RDF, OWL, RIF stored in file system (Semantic Web Stack)	Comprehensive but oriented to technical users	No	No	Good, electronic audit trail with provenance usable by business professionals	Manual or Automated

1.4.3. Problem solving logic

Computers work using the rules of mathematics. Mathematics works using the rules of logic. A problem solving logic⁴² is how a computer reasons.

To understand the notion of problem solving logic one first needs to understand the notion of logic and how logic can be applied to solving a problem. This section is dedicated to setting your perspective. The section provides specific definitions, deconstructing the pieces so that we can subsequently put the pieces back together.

The XBRL technical syntax is a global standard logic for representing knowledge. While much of the logic such as XBRL elements, relations between elements, mathematical relations between concepts and facts (XBRL calculation relations and XBRL Formula relations), dimensional relationships between concepts and facts, and other such relations (expressible using XBRL definition relations); not all such relation logic is standard.

XBRL Formula processors have specific deficiencies in their processing capabilities⁴³. To overcome these deficiencies, the following capabilities must exist or need to be added to XBRL Formula Processors:

- Support **normal global standard functionality** that high-quality XBRL Formula processors support (i.e. Arelle, UBmatrix/RR Donnelley, Fujitsu, Reporting Standards, etc.)

⁴¹ Business Logic System Gap Analysis, <http://xbrlsite.azurewebsites.net/2017/Library/BusinessLogicSystemGapAnalysis.jpg>

⁴² Comprehensive Introduction to Problem Solving Logic, http://xbrlsite.azurewebsites.net/2017/IntelligentDigitalFinancialReporting/Part01_Chapter02.5_ComprehensiveIntroductionToProblemSolvingLogic.pdf

⁴³ Specific Deficiencies in Capabilities of Existing XBRL Formula Processors, <http://xbrl.squarespace.com/journal/2016/9/26/specific-deficiencies-in-capabilities-of-existing-xbrl-formu.html>

- **Support inference** (i.e. deriving new facts from existing facts using logic, what inference engines do)
- Improved support validation and use of **structural relations** (i.e. XBRL Taxonomy functions; this was consciously left out of the XBRL Formula specification in order to focus on XBRL instance functionality)
- Support **forward chaining** and possibly also backward chaining in the future (i.e. chaining was also proposed but was left out of the XBRL Formula specification)
- Support a **maximum amount of Rulelog logic** which is safely implementable and is consistent with ISO/IEC Common Logic and OMG Semantics of Business Vocabulary and Business Rules
- **Additional XBRL definition arcroles** that are necessary to articulate the Rulelog logic, preferably these XBRL definition relation arcroles would end up in the XBRL International Link Role Registry and be supported consistently by all XBRL Formula processors (i.e. these general arcroles, and these financial disclosure related arcroles; this human readable information is helpful to understand the arcroles)

While added functionality might not be global standard functionality, the functionality is necessary to prove the logic of US GAAP based financial reporting or IFRS based financial reporting. US GAAP and IFRS semantics are relatively clear. What is not clear to some business professionals is how to represent that meaning using the XBRL global standard. Proprietary techniques for applying XBRL can be used to fill any gap. However, the logical rules used by any proprietary techniques should follow the logic of SCHACL, Common Logic, SBVR, and RuleLog.

1.4.4. Problem solving method

Computer systems use the information represented in some problem solving logic using some problem solving method or reasoning method. The problem solving method organizes and controls the steps taken, or line of reasoning, to solve a problem. There are two basic line of reasoning strategies: *forward chaining* and *backward chaining*.

Forward chaining reasoning starts with a set of conditions and moves toward some conclusion. **Backward chaining** reasoning starts with some known goal or conclusion and then creates a path to that conclusion.

I'm still trying to figure out the best processing approach for the fundamental accounting concept relations and other business rules to best take advantage of the power XBRL has to express business rules. One part of that question is whether a forward chaining or backward chaining approach is best.

The blog post *Forward and Backward Chaining: Part 2*⁴⁴, by Charles Forgy, PhD, provided the best answer to that question that I have come across. In a nutshell, here is Dr. Forgy's answer: use both if possible:

In conclusion, forward and backward chaining systems both use subgoals to control the operation of a rule base. Pure forward chaining systems are more powerful than pure backward chaining systems, but pure forward chaining systems require the

⁴⁴ Charles Forgy, PhD, *Forward and Backward Chaining: Part 2*, http://web.archive.org/web/20050904233822/http://www.rulespower.com/forgy_chaining_2.htm

developer to write all the subgoaling rules. Modern forward chaining systems such as the RulesPower system integrate automatic backward chaining with forward chaining. These systems combine the best features of both forward and backward chaining.

Dr. Forgy distills the difference between forward and backward chaining down to the following two salient points:

- Rule firing:
 - 1. Forward-chaining systems fire rules whenever the rules' If parts are satisfied.
 - 2. Backward-chaining systems attempt to fire rules only when those rules can potentially satisfy a goal.
- Subgoal creation:
 - 1. Backward-chaining systems automatically create new subgoals when more information is needed to determine whether a given rule is satisfied.
 - 2. Forward-chaining systems do not automatically create subgoals.

Dr. Forgy explains why you would want to use forward-chaining when backward-chaining automatically creates new subgoals but forward-chaining does not:

You might ask why you would want to use a forward chaining system if you have to write rules to manage subgoals. After all, backward chaining systems automatically manage the subgoals. There answer is very simple: **Backward chaining systems are more limited than forward chaining systems**. There are many kinds of tasks that can be handled easily with a forward chaining system that are either difficult or impossible with a backward chaining system. Backward chaining systems are good for diagnostic and classification tasks, but they are not good for planning, design, process monitoring, and quite a few other tasks. Forward chaining systems can handle all these tasks.

So there seems to be a tradeoff. You can use forward-chaining and satisfy all your needs, but the price you pay is having to create subgoals.

Alternatively, you could use a system that supports forward-chaining which automatically switches to a backward-chaining mode when needed.

If you want to understand more details about forward and backward chaining, Dr. Forgy first part in this series, Forward and Backward Chaining Part 1, is worth reading.

But man, why can't this be easy! Note this statement in the last paragraph on the Wikipedia article about business rule engine types:

A fourth class of rules engine might be called a deterministic engine. These rules engines may forgo both forward chaining and backward chaining, and instead utilize domain-specific language approaches to better describe policy. This approach is often easier to implement and maintain, and provides performance advantages over forward or backward chaining systems.

When a forward chaining approach is used, the rules are traversed from the problem to the solution to the problem.

- If A then B

- If B then C
- If C then D

A chain traversed from a hypothesis back to the facts that support the hypothesis is a backward chain.

- If D then C
- If C then B
- If B then A

1.4.5. When to use a Rules Engine

Rules engines are not a panacea. Ultimately, the logic has to end up somewhere, either embedded in software or in a rules engine. Each approach has pros and cons. This article, *Some Guidelines For Deciding Whether To Use A Rules Engine*⁴⁵, by George Rudolph, offers guidance to help you understand when and when not to use a rules engine with your rules.

1.5. Scaling Business Rules

Managing business rules becomes more complex as the number of rules increase. Scaling business rules is important. Using a decision model based approach can help manage large sets of business rules. The article, *How DMN Allows Business Rules to Scale*⁴⁶ points out four primary problems that you run into:

- **The 'Rush to Detail'**: business rule development encourages policy makers to focus on rule implementation prematurely, before they have considered the broader goals and structure of their business decisions and to what extent they will be automated. This approach is like starting to build a house by laying bricks, rather than drawing plans and establishing foundations.
- **Poor Dependency Management**: a growing and poorly understood set of inter-dependencies between rules causing changes to have unintended consequences—making the rule set brittle and reducing its agility.
- **Insufficient Transparency**: the bewildering size of a rule set, use of technical (rather than business) terms and style for expressing rules and a poor connection between rules and their business context (their rationale and place in the business process)—making the meaning and motivation of rules more obscure.
- **Lack of Growth Management**: poor discipline about the scope, quality and placement of rules that are added to the rule set—making it hard to find rules and leading to 'stale' rules and duplicates.

⁴⁵ George Rudolph, *Some Guidelines For Deciding Whether To Use A Rules Engine*, <http://www.jessrules.com/guidelines.shtml>

⁴⁶ Jan Purchase, *How DMN Allows Business Rules to Scale*, November 19, 2015, <http://blog.luxmagi.com/2015/11/how-dmn-allows-business-rules-to-scale/>

1.6. Constrained Natural Language Business Rules Syntax

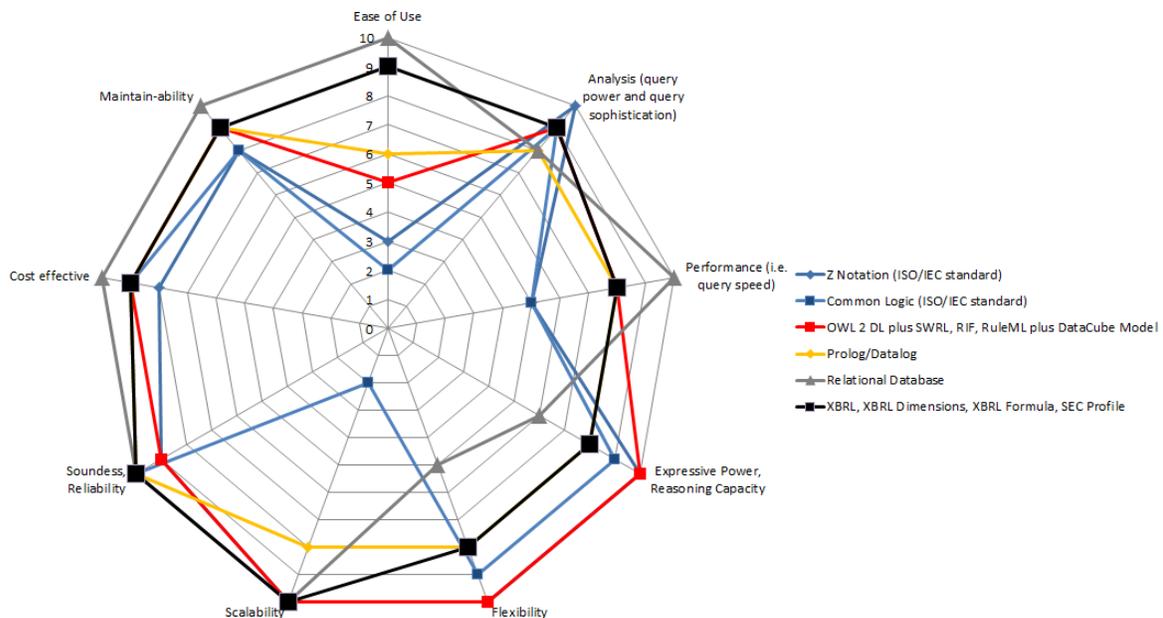
There are all sorts of syntaxes for business rules. Natural language is one syntax or format. By constrained natural language we mean: (a) Only words from a specified, limited set of words is allowed; (b) only specified forms of phrase and clause is used; and (c) words, phrases, and clauses are only used in specified combinations.

The benefit of constrained natural language business rules is that they are easy for business professionals to read. A properly created constrained natural language business rules set can be converted to any other syntax using machine-based processes.

Here is guidance for writing quality natural language business rules: <https://www.brcommunity.com/articles.php?id=b660>

1.7. Achieving Balance and Equilibrium

Every solution is a basket of pros and cons. The best solution depends on the requirements of the problem that is being solved. And so understanding your requirements correctly is paramount when it comes to being sure you select the right approach to solving any problem.



The list of requirements that I have come up with include the following:

- **Ease of use:** How easy is the system for business professionals to use?
- **Analysis (query power and query sophistication):** When you are gathering information to make use of that information, how sophisticated are the queries?
- **Performance (query speed):** What is the raw speed at which information is provided for use?

- **Expressive power, reasoning capacity:** What is the expressive power of the business rules? There is a direct correlation between expressive power and the reasoning capacity the system can offer.
- **Flexibility:** Is the system flexible enough to meet your needs?
- **Scalability:** Will the system scale if your needs grow?
- **Soundness, reliability:** Is the system reliable and sound?
- **Cost effectiveness:** What is the cost of the system relative to what the system provides?

1.8. More Background Information on Business Rules

In the 1980s, the original 'white paper' on business rules was published, *Defining Business Rules ~ What Are They Really?*⁴⁷ That document contains a lot of very useful information about business rules.

The business rules mantra⁴⁸ states:

"Rules are a first-class citizen of the requirements world."

"Rules build on facts, and facts build on concepts as expressed by terms."

The business rules manifesto⁴⁹ is worth reading.

⁴⁷ David Hay et.al., *Defining Business Rules ~ What Are They Really?*, https://www.businessrulesgroup.org/first_paper/BRG-whatBR_3ed.pdf

⁴⁸ Business rules group, *Business Rules Mantra*, https://www.businessrulesgroup.org/brapproach_mantra.htm

⁴⁹ Business rules group, *Business Rules Manifesto*, <https://www.businessrulesgroup.org/brmanifesto.htm>